

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for generating a Concrete Model of a computing utility comprising the steps of:

receiving as input an infrastructure-independent Service Environment Model of a service environment, said Service Environment Model describing a set of requirements for an initial desired state of said service environment;

receiving as input an Infrastructure Model describing both hardware and software resources and an organization of said resources in the computing utility infrastructure, said Infrastructure Model representing knowledge encapsulated in a knowledge subsystem, wherein said knowledge subsystem comprises a set of objects used to represent resource instances and relationships, configure resources and relationships, query their state, and query their configuration capabilities and constraints; and

generating provisioning actions to reach a state that satisfies the set of requirements specified in the Service Environment Model, wherein the generating step comprises steps of:

merging the Service Environment Model with the Infrastructure Model to generate the Concrete Model, said Concrete Model describing a structure to implement on the computing utility infrastructure in order to reach the desired state as expressed in the Service Environment Model and being mappable to said knowledge subsystem, said generating comprising steps of:

executing ~~at least one~~ a plurality of refinement steps, each refinement step comprising:

selecting a node;

wherein nodes represent resources and requirements on a state of the resources; and

wherein edges represent relationships between the resources; and

replacing the selected node with a sub graph structure to obtain an intermediary model which is an input to a subsequent refinement step;

iteratively refining the intermediary models produced by each replacing step ~~model~~ by repeating the steps of selecting and replacing until a resulting intermediary model is determined to be mappable to the knowledge subsystem, wherein the resulting intermediary model is determined to be mappable when every node in the resulting intermediary model ~~Concrete Model~~ that represents a resource is mappable to the knowledge subsystem;
halting the refining when no further refinements can be made; and
mapping the resulting intermediary model to the knowledge subsystem.

2. (Previously presented) A method as recited in claim 1, wherein the step of receiving as input the Service Environment Model of the service environment comprises receiving a description of a set of requirements on another desired state of said service environment.

3. (Canceled)

4. (Original) A method as recited in claim 1, wherein said service environment is an entity taken from a group of entities consisting of: a Web site, an on-line gaming service, a scientific computation service, an e-business service, a computing service, and any combination of these.

5. (Previously presented) A method as recited in claim 1, implemented as an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing generation of a Concrete Model, the computer readable program code means in said article of manufacture comprising code for causing a computer to effect the steps of claim 1.

6. (Previously presented) A method as recited in claim 1, wherein the step of receiving as input the Infrastructure Model comprises an action taken from a group of actions consisting of:

querying at least one knowledge subsystem entity; querying Resource Managers, querying Resource Instance Services, querying a best practices catalog; obtaining knowledge of available resource types; obtaining knowledge of resources constraints; obtaining knowledge of resource capabilities; obtaining knowledge of infrastructure constraints; obtaining knowledge of infrastructure capabilities; obtaining knowledge of infrastructure best practices patterns; and any combination of these actions.

7. (Canceled)

8. (Previously presented) A method as recited in claim 1 wherein the step of replacing comprises at least one limitation selected from a group of limitations consisting of: querying a best practices catalog; generating sub graph patterns dynamically; employing graph matching techniques to obtain said sub-graph structure; and employing graph merging techniques to obtain said sub-graph structure.

9. (Previously presented) A method as recited in claim 1, implemented as a program storage device readable by computer, tangibly embodying a program of instructions executable by the computer to perform method steps for generating the Concrete Model, said method steps comprising the steps of claim 1.

10. (Previously presented) A method as recited in claim 1, further comprising using the Concrete Model to enforce a policy based service provider's best practices in implementation of Service Environments in the computing utility infrastructure.

11. (Previously presented) A method as recited in claim 10, wherein the best practices are encoded as patterns in a best practices catalog and used in the step of generating the Concrete Model.

12. (Previously presented) A method as recited in claim 1, further comprising employing said Concrete Model to generate provisioning actions, said provisioning actions, when executed, create a resource structure that matches the description in the Concrete Model; and
said resource structure satisfying said set of requirements on new desired state of said service environment.

13. (Original) A method as recited in claim 12, further comprising employing said provisioning to enforce a policy based service provider's best practices in implementation of service environments in the computing utility infrastructure.

14. (Previously presented) A method as recited in claim 13, wherein the best practices are encoded as patterns in a best practices catalog and used in the step of generating the Concrete Model.

15. (Original) A method as recited in claim 12, wherein step of provisioning includes a task taken from a group of tasks consisting of: creating a new service environment, changing the combination of resources allocated to a service environment, changing the configuration of resources allocated to a service environment, or destroying a service environment, or any combination of the above.

16. (Previously presented) A method as recited in claim 15, wherein changing the configuration of resources allocated to a service environment comprises changing a local state of a resource or changing the way the resource is configured to work with other resources.

17. (Previously presented) A method as recited in claim 1, wherein the method is used to regenerate provisioning instructions whenever at least one of the following occurs: infrastructure characteristics change; and requirements of a service change.

18. (Previously presented) A method as recited in claim 17, wherein the infrastructure characteristics comprise a at least one characteristic taken from a group of characteristics consisting of: types of resources in the infrastructure; capabilities of said resources; configuration of said resources; constraints on configuration of said resources; and best practices patterns as defined in the best practices catalog.

19. (Previously presented) A method as recited in claim 1, further comprising employing said Concrete Model to generate a Resource Manager for a composite resource, and provisioning and managing computing services in a computing utility system, based on a high level description of the characteristics and structure of desired computing services and a representation of the computing utility infrastructure used as a platform to implement the said computing services.

20. (Previously presented) A method as recited in claim 19, wherein said Resource Manager provides a set of resource manager methods taken from a group of resource manager methods consisting of: creating composite resources based on a Concrete Model; changing composite resources based on a Concrete Model; destroying composite resources based on a Concrete Model; or any combination of these methods.

21. (Currently amended) An apparatus comprising means for generating a Concrete Model, said Concrete Model describing a structure of resources implementable over a computing utility infrastructure, and satisfying a set of service requirements, said means for generating comprising the steps of:

obtaining a Service Environment Model of a service environment, said Service Environment Model describing a desired state of said service environment;

obtaining an Infrastructure Model describing both hardware and software resources and an organization of the resources in the computing utility infrastructure, said Infrastructure Model representing knowledge is encapsulated in a knowledge subsystem, wherein said knowledge subsystem comprises a set of objects used to represent resource instances and relationships,

configure resources and relationships, query their state, and query their configuration capabilities and constraints; and

generating provisioning actions to reach a state that satisfies the set of requirements specified in the Service Environment Model, wherein the generating step comprises steps of:

merging the Service Environment Model with the Infrastructure Model to generate a Concrete Model describing a resource structure such that said Concrete Model refines the Service Environment Model and is mappable to said knowledge subsystem, said generating comprising steps of:

executing a plurality of refinement steps, each refinement step comprising:

selecting a node;

wherein nodes represent resources and requirements on a state of the resources; and

wherein edges represent relationships between the resources; and

replacing the selected node with a sub graph structure to obtain an intermediary model which is an input to a subsequent refinement step;

iteratively refining the intermediary models produced by each replacing step by repeating the steps of selecting and replacing until a resulting intermediary model is determined to be mappable to the knowledge subsystem, wherein the resulting intermediary model is determined to be mappable when every node in the resulting intermediary model that represents a resource is mappable to the knowledge subsystem;

halting the refining when no further refinements can be made; and

mapping the resulting intermediary model ~~Concrete Model~~ to the knowledge subsystem.

22. (Previously presented) An apparatus as recited in claim 21, further comprising means for employing said Concrete Model to generate provisioning actions, said provisioning actions, when executed, create a resource structure that matches the description in the Concrete Model, said resource structure satisfies said desired state of said service environment.

23. (Previously presented) An apparatus as recited in claim 21, implemented as a computer program product comprising a computer usable medium having computer readable program code means embodied therein for causing generation the Concrete Model, the computer readable program code means in said computer program product comprising code for causing a computer to effect the functions of claim 21.

24. (Original) An apparatus as recited in claim 21, further comprising means for employing said Concrete Model to generate a Resource Manager for a composite resource.

25. (Previously presented) A method as recited in claim 1, where the step of generating a Concrete Model is performed by a user taken from a group of users consisting of: a service provider; a customer of a service provider; a company owning an IT infrastructure; and a utility provider.